Sprawozdanie z laboratorium Inżynieria Oprogramowania

Semestr letni 2016/2017

Metodyka – Extreme programming

|  |  |
| --- | --- |
| Wersja: **1.1** | Data: **14.03.2017** |
| Opis zmian   * dodano założenia projektu | |
| Wersja: **1.2** | Data: **14.03.2017** |
| Opis zmian   * dodano ograniczenia systemowe | |
| Wersja: **1.3** | Data: **14.03.2017** |
| Opis zmian   * dodano specyfikacje wymagań | |
| Wersja: **1.4** | Data: **14.03.2017** |
| Opis zmian   * dodano spis wymagań | |
| Wersja: **2.1** | Data: **21.03.2017** |
| * wykonano diagram Rich Picture | |
| Wersja: **2.2** | Data: **21.03.2017** |
| * napisano komentarz do diagramu | |
| Wersja: **3.1** | Data: **3.04.2017** |
| * wykonano diagram przypadków użycia wraz z komentarzem | |
| Wersja: **3.2** | Data: **4.04.2017** |
| * dodano opis aktorów | |
| Wersja: **3.3** | Data: **4.04.2017** |
| * dodano opis przypadków | |
| Wersja: **4.1** | Data: **8.05.2017** |
| * wykonano diagram podsystemów z komentarzem | |
| Wersja: **4.2** | Data: **8.05.2017** |
| * dodano opis podsystemów | |

Spis treści

[**Analiza wymagań**](#_1fob9te) **3**

[Założenia projektowe](#_3znysh7) 3

[Ograniczenia systemu](#_2et92p0) 3

[Specyfika działania](#_tyjcwt) 3

[Spis wymagań](#_3dy6vkm) 3

[**Rich picture**](#_1t3h5sf) **3**

[Diagram](#_4d34og8) 3

[Komentarz](#_2s8eyo1) 3

[**Przypadki użycia**](#_17dp8vu) **3**

[Diagram wraz z komentarzem](#_3rdcrjn) 3

[Opis aktorów](#_26in1rg) 3

[Opis przypadków](#_lnxbz9) 3

[**Diagram podsystemów**](#_35nkun2) **4**

[Grafika wraz z komentarzem](#_1ksv4uv) 4

[Opis podsystemów](#_44sinio) 4

[**Wybrane diagramy aktywności**](#_2jxsxqh) **5**

[Diagram aktywności: “<nazwa>” wraz z komentarzem](#_z337ya) 5

[Diagram aktywności: “<nazwa>” wraz z komentarzem](#_3j2qqm3) 5

[Diagram aktywności: “<nazwa>” wraz z komentarzem](#_1y810tw) 5

[Diagram aktywności: “<nazwa>” wraz z komentarzem](#_4i7ojhp) 5

[**Diagram klas**](#_2xcytpi) **5**

[Grafika wraz z komentarzem](#_1ci93xb) 5

[Opis klas](#_3whwml4) 5

[**Diagramy sekwencji**](#_2bn6wsx) **6**

[Diagram sekwencji: “<nazwa>”](#_qsh70q) 6

[Diagram sekwencji: “<nazwa>”](#_3as4poj) 6

[Diagram sekwencji: “<nazwa>”](#_1pxezwc) 6

[**Diagram rozmieszczenia na procesorach**](#_49x2ik5) **6**

[Grafika wraz z komentarzem](#_2p2csry) 6

[**Diagramy stanów**](#_147n2zr) **6**

[Diagram stanów: “<nazwa>”](#_3o7alnk) 6

[Diagram stanów: “<nazwa>”](#_23ckvvd) 6

[**Lista katalogów i plików wchodzących w skład projektu**](#_ihv636) **6**

# Analiza wymagań

## Założenia projektowe

* Pracownicy potrafią obsługiwać system komputerowy
* Specjaliści poszczególnych działów są przeszkoleni i wiedzą jakie są ich obowiązki
* Na terenie firmy znajduje się zasilanie awaryjne
* Rury hydraulicznie nie mogą znajdować się pod sufitem
* Firma posiada serwer bazodanowy
* Pracownicy wyposażeni są w karty RFID
* Firma posiada aktywny system ochrony i kamer
* Liczba bramek/`wejść na teren firmy wynosi 5
* Każda karta posiada dane personalne pracownika

## Ograniczenia systemu

* Liczba użytkowników jest ograniczona do 1000
* Każdy dział ma swoje indywidualne uprawnienia do modyfikacji i odczytu (np. szef bezpieczeństwa ma możliwość zarządzania uprawnieniami kart pracowników)
* Karta rejestruje wyłącznie operacje wejścia/wyjścia na terenie firmy
* Każdy pracownik posiada dokładnie jedną kartę RFID
* System obsługuje tylko karty magnetyczne, które zostały wydane przez firmę

## Specyfika działania

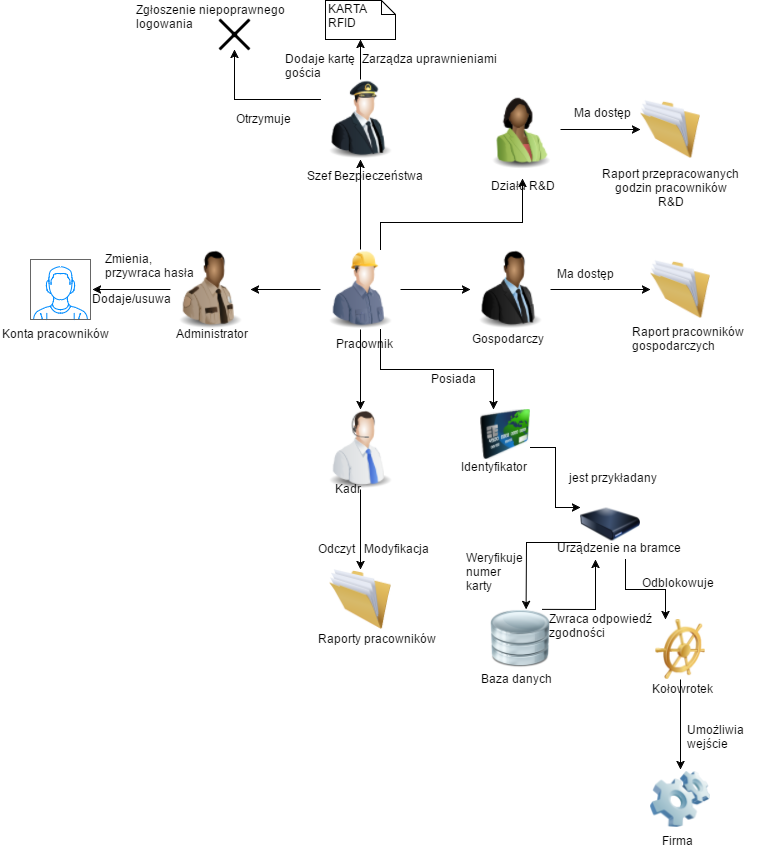
* Urządzenie przy wejściu weryfikuje dane pracownika poprzez sprawdzenie jego numeru karty w bazie danych
* Po poprawnej weryfikacji danych następuje odblokowanie kołowrotka
* Przy awarii prądu system nadal działa z wykorzystaniem zasilania awaryjnego (UPS)
* Każdy dział posiada swój unikalny interfejs graficzny, na którym może wykonywać tylko i wyłącznie operacje jakie jego dział ma w swoich preferencjach
* Szyfrowanie danych pracowników jest wykonywane za pomocą TDE (Transparent Data Encryption)
* System okresowo nakazuje zmianę hasła zgodną z założeniami Microsoft’u dotyczącymi polityki haseł
* Wymagana jest okresowa zmiana hasła

## Spis wymagań

1. Urządzenie przy każdym wejściu jest wyposażony w system rejestracji wejścia/wyjścia użytkowników
2. Urządzenie musi odblokować pracownikowi kołowrotek znajdujący się przy wejściu
3. System działa w przypadku awarii prądu za pomocą zasilacza UPS. Zasilacz wyposażony jest w akumulator, i w przypadku przerwy lub zakłóceń dostawy energii elektrycznej z sieci energetycznej system przełącza się na pracę z akumulatora. Czas podtrzymania napięcia wynosi do kilkudziesięciu godzin
4. System rejestruje prywatne i służbowe wyjścia podczas pracy
5. Pracownicy działu R&D mają możliwość dostępu wyłącznie do swojego raportu przepracowanych godzin ze swojego komputera
6. Pracownicy gospodarczy mają możliwość dostępu wyłącznie do własnego raportu z poziomu ogólnodostępnego terminala
7. Pracownicy kadr mają dostęp do raportu wszystkich pracowników z możliwościa odczytu i edycji wpisów
8. Szef bezpieczeństwa mają możliwość zarządzania uprawnieniami kart RFID oraz możliwość dodania karty gościa
9. Szef bezpieczeństwa odbiera zgłoszenie w przypadku niepoprawnego logowania do systemu w postaci komunikatu informującego o błędzie na swoim stanowisku komputerowym
10. Pracownicy mają możliwość logowania się za pomocą loginu i hasła
11. Administrator dodaje, usuwa konta pracowników oraz zmienia i przywraca ich hasła

# Rich picture

## Diagram



## Komentarz

Głównym naszym polem jest Pracownik, który jest ogółem wszystkich osób zatrudnionych w firmie, dzieli się na poszczególne stanowiska oraz przysługuje mu Identyfikator. Każde ze stanowisk posiada indywidualne atrybuty, co skutkuje rozszerzeniem drzewa Rich Picture. Drugim najważniejszym elementem jest Urządzenie na bramce. Od tego obiektu wykonuje się kilka operacji umożliwiających wstęp do firmy. Dzięki metodzie Rich Picture mamy uogólniony model projektu, który jest zrozumiały dla klienta. -

# Przypadki użycia

## Diagram wraz z komentarzem



Model przypadków użycia dostarcza abstrakcyjnego spojrzenia na system - spojrzenia z pozycji aktorów, którzy go używają. Nie włącza zbyt wielu szczegółów, co pozwala wnioskować o funkcjonalności systemu na odpowiednio wysokim poziomie. Jest przypadkiem, w którym dany system jest używany w celu spełniania jednego lub większej liczby wymagań użytkowników. Wychwytuje fragment funkcji udostępnianych przez system. Określają wymagania funkcjonalne systemu. Aktorzy mogą być ludźmi wchodzącymi w interakcję, systemami zewnętrznymi.

## Opis aktorów

|  |  |
| --- | --- |
| Nazwa aktora | Opis |
| Pracownik | Dowolna osoba ,z uprawnieniami do wejścia na teren firmy, która jest w systemie oznaczona jako zatrudniona. Aktor ożywiony, zewnętrzny |
| Pracownik gospodarczy | Osoba posiadająca dostęp do własnego raportu.  Aktor ożywiony, zewnętrzny |
| Pracownik działu R&D | Osoba posiadająca dostęp do własnego raportu  Aktor ożywiony, zewnętrzny |
| Pracownik kadr | osoba mogąca odczytywać i modyfikować raporty pracowników  Aktor ożywiony, zewnętrzny |
| Szef działu bezpieczeństwa | osoba zarządzająca uprawnieniami, dodająca karty gości oraz otrzymująca alert o niepoprawnym logowaniu  Aktor ożywiony, zewnętrzny |
| Administrator | osoba zarządzająca kontami użytkowników firmy (dodawanie, usuwanie, zmiana haseł)  Aktor ożywiony, zewnętrzny |
| Baza danych | Magazynuje dane takie jak raporty, informacje na temat kart pracowników, kart gości oraz informacje o ich uprawnieniach. |
| Baza kont pracowników | Zawiera informacje na temat kont pracowników. |
| System kontroli bramek | Odpowiedzialny za otwieranie bramek. |

## Opis przypadków

|  |  |
| --- | --- |
| Nazwa UC | Wejscie przez bramke |
| Skrócony opis UC | Pozwala wejść na teren firmy |
| Aktor/Aktorzy | Pracownik |
| Warunki początkowe | 1. Pracownik posiada karte 2. Pracownik posiada konto |
| Zdarzenie inicjujące | Przyłożenie karty do bramki |
| Przebieg | 1. Rejestracja próby wejścia do bazy danych 2. Sprawdzenie zgodności karty RFID w bazie danych 3. Odblokowanie bramki |
| Przebiegi alternatywne | 4. Bramka nie otwiera się.  5. Wysłanie alertu o błednej próbie wejścia do szefa bezpieczeństwa |
| Warunki końcowe | Wejście na teren firmy |
| Powiązane UC | <<include>> rejestrowanie wejścia  <<include>> sprawdzenie zgodności karty RFID |

|  |  |
| --- | --- |
| Nazwa UC | Logowanie |
| Skrócony opis UC | Umożliwia wejście na konto pracownika |
| Aktor/Aktorzy | Pracownik |
| Warunki początkowe | 1. Pracownik posiada konto w systemie |
| Zdarzenie inicjujące | 1. Uruchomienie stanowiska komputerowego |
| Przebieg | 1. Wpisanie loginu oraz hasla w panelu logowania 2. Poinformowanie o logowaniu do bazy kont pracowników |
| Przebiegi alternatywne | 1. Alert o blednej autoryzacji danych podanych do konta logowania pracownika |
| Warunki końcowe | Poprawne zalogowanie na swoje konto |
| Powiązane UC | <<extend>> zgłoś alert niepoprawności |

|  |  |
| --- | --- |
| Nazwa UC | Odczytaj raport |
| Skrócony opis UC | Umożliwia odczytanie własnego raportu |
| Aktor/Aktorzy | Pracownik |
| Warunki początkowe | 1. Pracownik posiada konto w systemie |
| Zdarzenie inicjujące | 1. Poprawne zalogowanie na konto |
| Przebieg | 1. Pracownik za pośrednictwem GUI wysyła prośbę do bazy danych o otrzymanie raportu do wglądu 2. Otrzymanie raportu 3. Odczytanie raportu |
| Przebiegi alternatywne | 1. Brak odpowiedzi z bazy danych |
| Warunki końcowe | Odczytanie własnego raportu |
| Powiązane UC | <<include>> zapisuj aktywność |

|  |  |
| --- | --- |
| Nazwa UC | Modyfikuj raport |
| Skrócony opis UC | Umożliwia wprowadzanie zmian do raportów |
| Aktor/Aktorzy | Pracownik kadr |
| Warunki początkowe | 1. Konto w systemie 2. Uprawnienia dotyczące modyfikacji |
| Zdarzenie inicjujące | 1. Poprawne zalogowanie na konto |
| Przebieg | 1. Pracownik za pośrednictwem GUI wysyła prośbę do bazy danych o otrzymanie raportu do wglądu 2. Otrzymanie raportu 3. Modyfikacja raportu 4. Zapisanie zmian i wysłanie zmodyfikowanego raportu do bazy danych |
| Przebiegi alternatywne | 1. Brak odpowiedzi z bazy danych |
| Warunki końcowe | Modyfikacja danego raportu |
| Powiązane UC | <<include>> zapisuj aktywność |

|  |  |
| --- | --- |
| Nazwa UC | Dodawaj konta |
| Skrócony opis UC | Umożliwia administratorowi dodawanie kont pracowników |
| Aktor/Aktorzy | Administrator |
| Warunki początkowe | 1. Dostęp do bazy kont pracowników |
| Zdarzenie inicjujące | 1. Zalogowanie sie do panelu administracyjnego |
| Przebieg | 1. Logowanie do panelu administracyjnego 2. Wejście w zakladke “dodaj konto”. 3. Podanie danych pracownika oraz ustawienie dla niego hasła 4. Zapisanie zmian |
| Przebiegi alternatywne | 5. Alert o juz istniejącym koncie |
| Warunki końcowe | Utworzenie nowego konta |
| Powiązane UC | <<include>> generuj hasła |

|  |  |
| --- | --- |
| Nazwa UC | Usuwaj konta |
| Skrócony opis UC | Umożliwia usuwanie konta |
| Aktor/Aktorzy | Administrator |
| Warunki początkowe | 1. Dostęp do bazy kont pracowników |
| Zdarzenie inicjujące | 1. Poprawne zalogowanie do panelu administracyjnego |
| Przebieg | 1. Wejście w zakładkę “usuń konto”. 2. Wybór konta do usunięcia 3. Zapisanie zmian |
| Przebiegi alternatywne | 2. Lista kont jest pusta, baza kont pracowników nie odpowiada |
| Warunki końcowe | Usunięcie wybranego konta |
| Powiązane UC | brak powiązań |

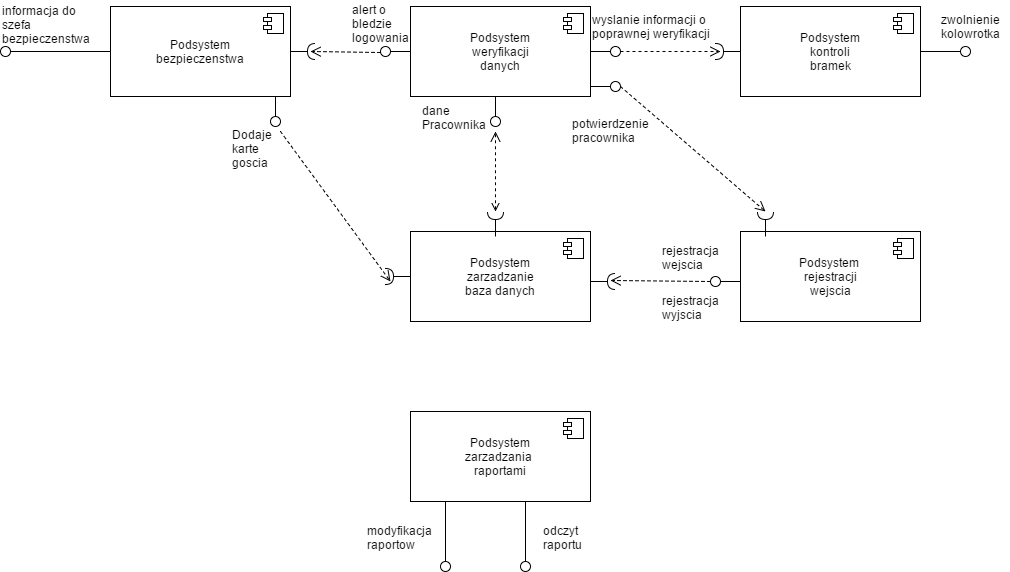
|  |  |
| --- | --- |
| Nazwa UC | Dodanie karty gościa |
| Skrócony opis UC | Umożliwia dodanie karty gościa do systemu |
| Aktor/Aktorzy | Szef bezpieczeństwa |
| Warunki początkowe | 1. Dostęp do bazy danych ze swojego stanowiska |
| Zdarzenie inicjujące | 1. Wydanie prośby o kartę gościa |
| Przebieg | 1. Prośba o dokument tożsamości 2. Logowanie do systemu 3. Wpisanie danych do systemu 4. Zatwierdzenie zmian |
| Przebiegi alternatywne | 2. Błędne logowanie |
| Warunki końcowe | Wydanie karty gościa |
| Powiązane UC | brak powiązań |

|  |  |
| --- | --- |
| Nazwa UC | Odblokuj bramke |
| Skrócony opis UC | Powoduje otworzenie bramki |
| Aktor/Aktorzy | System kontroli bramek |
| Warunki początkowe | 1. Poprawna weryfikacji karty z bazą danych |
| Zdarzenie inicjujące | 1. Otrzymanie zgody na otworzenie bramki |
| Przebieg | 1. Otrzymanie poprawnej odpowiedzi z bazy danych 2. Zwolnienie blokady |
| Przebiegi alternatywne | 1. Otrzymanie negatywnej odpowiedzi z bazy danych |
| Warunki końcowe | Otworzenie bramki |
| Powiązane UC | <<include>>sprawdz zgodnosc karty RFID |

|  |  |
| --- | --- |
| Nazwa UC | Zarządzanie uprawnieniami kart RFID |
| Skrócony opis UC | Odpowiada za przydzielenie i odbior uprawnien dla kart pracowniczych RFID |
| Aktor/Aktorzy | Szef bezpieczenstwa |
| Warunki początkowe | 1. Dostep do bazy danych pracownikow. |
| Zdarzenie inicjujące | 1. Informacja na temat zmiany uprawnien okreslonego pracownika |
| Przebieg | 1. Zalogowanie sie do bazy danych pracownikow ze swojego terminala 2. Dodanie/Odebranie uprawnien 3. Zapisanie zmian |
| Przebiegi alternatywne | 1. Niedziałający terminal |
| Warunki końcowe | Przypisanie odpowiednich uprawnien |
| Powiązane UC | brak powiazań |

# Diagram podsystemów

## Grafika wraz z komentarzem



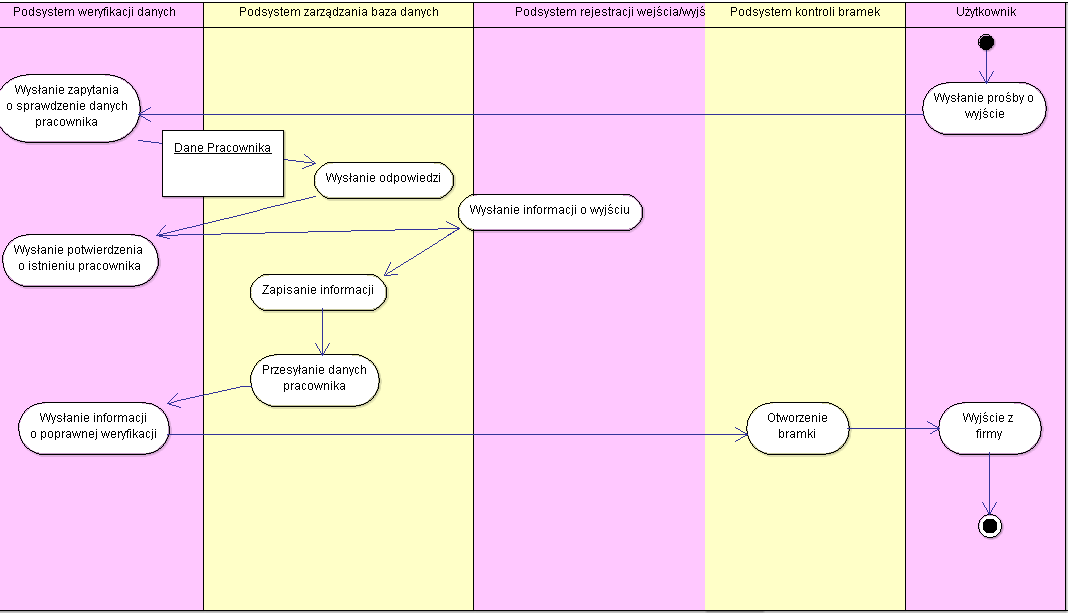
Grafika diagramu podsystemów przedstawia rolę podsystemów w całym systemie oraz związki lub interfejsy pomiędzy poszczególnymi podsystemami. Funkcje w systemie logicznym muszą być połączone do zobrazowania komunikacji. W Diagramie podsystemów wykorzystujemy 2 rodzaje łączników. Pierwszy z nich(pół kolisty) należy do grupy interfejsów wymagaych. Natomiast drugi typ(kołowy) należy do grupy interfejsów dostarczających, czyli takich, które są wymagane do funkcjonowania komponentu.

## Opis podsystemów

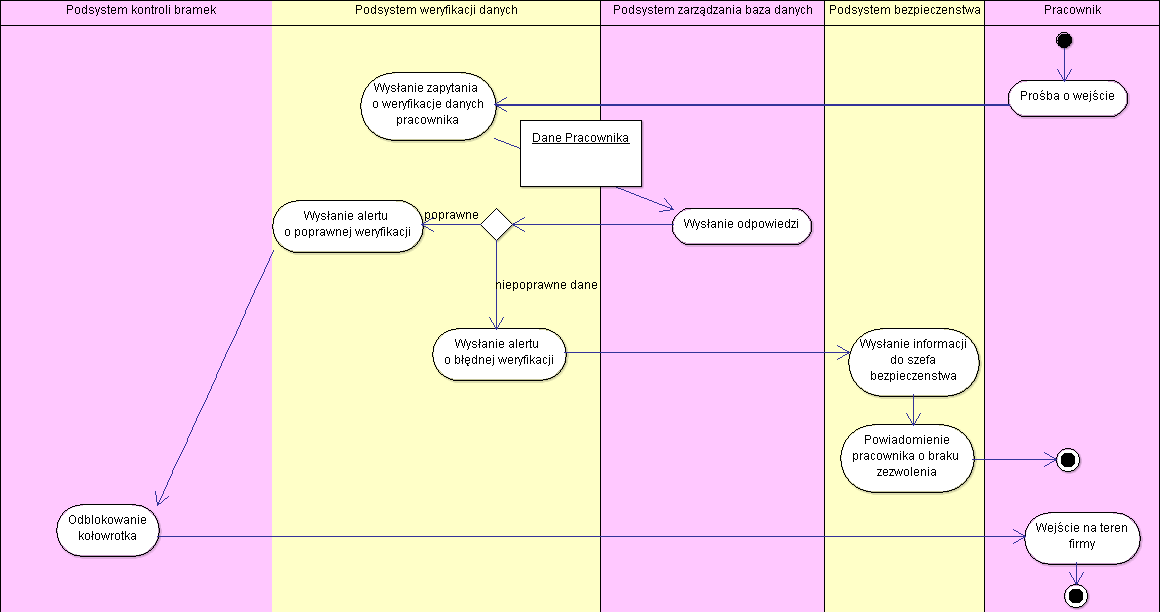
|  |  |
| --- | --- |
| Nazwa podsystemu | Opis |
| Podsystem weryfikacji danych | Wysyła alert o błędzie oraz poprawnej weryfikacji. Weryfikuje pracownika i wysyła potwierdzenie do rejestracji |
| Podsystem zarządzania bazą danych | Przesyła odpowiedzi zgodności danych pracownika |
| Podsystem bezpieczeństwa | Otrzymuje zgłoszenie o niepoprawnej weryfikacji i wysyła informację do szefa bezpieczeństwa. Dodaje kartę gościa. |
| Podsystem kontroli bramek | Zwalnia kołowrotek |
| Podsystem rejestracji wejścia | Przesyła informacje o wejściu/wyjściu na teren firmy do bazy danych. |
| Podsystem zarządzania raportami | Odczyty raportu, możliwość modyfikacji raportu. |

# Wybrane diagramy aktywności

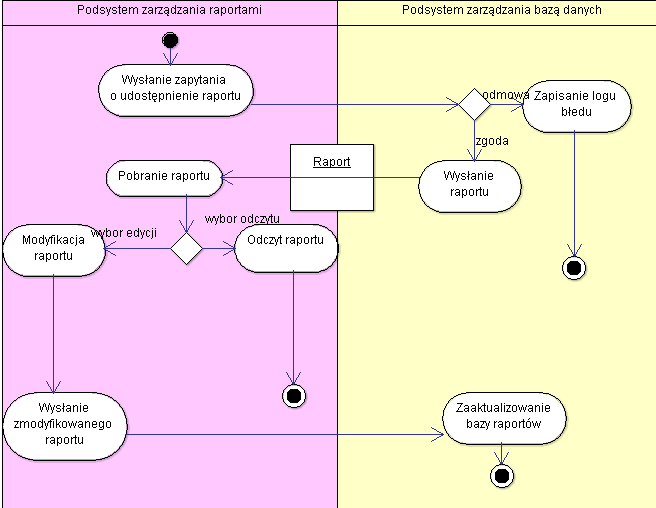
## Procedura rejestrowania wyjścia służbowego pracownika wraz z otworzeniem bramki



## Procedura wejścia pracownika na teren firmy



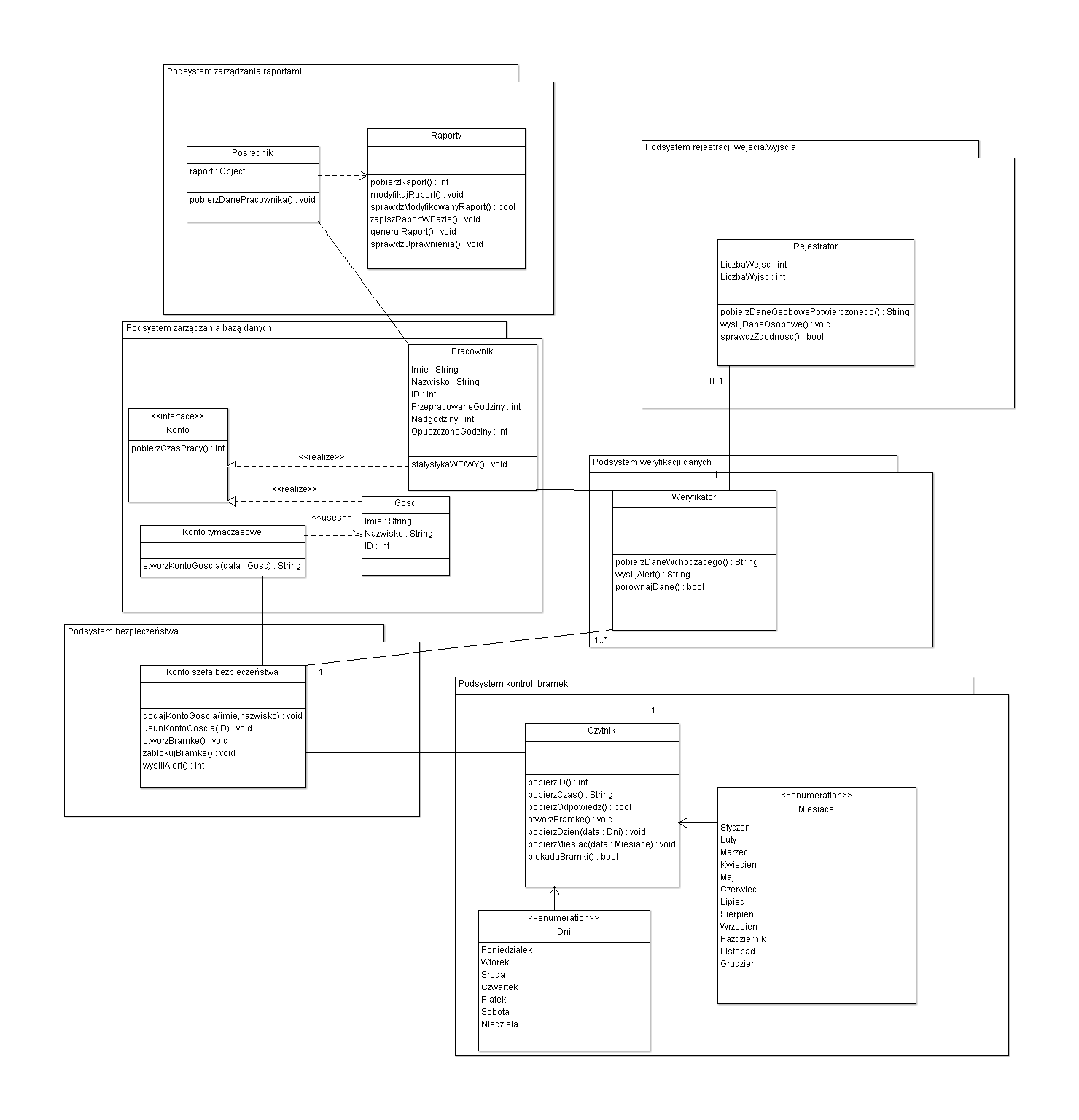
## Procedura odczytu i modyfikacji raportu



## Procedura dodania karty goscia

# Diagram klas

## Grafika wraz z komentarzem

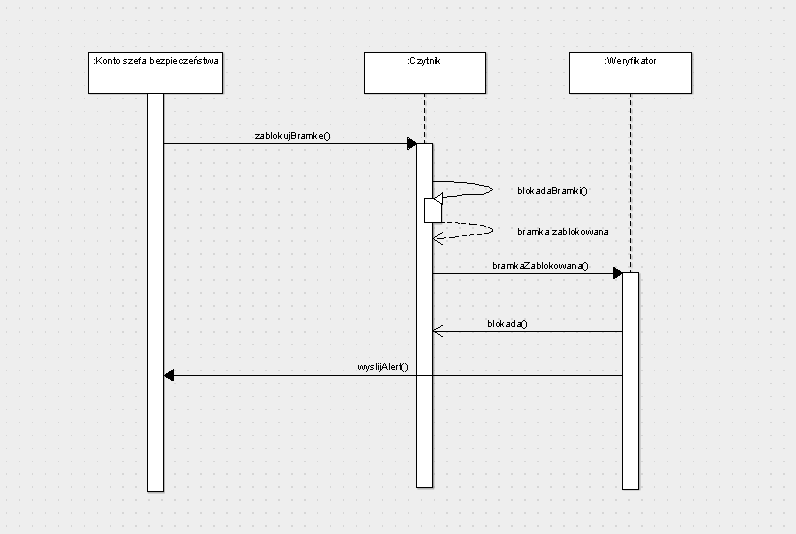


## Opis klas

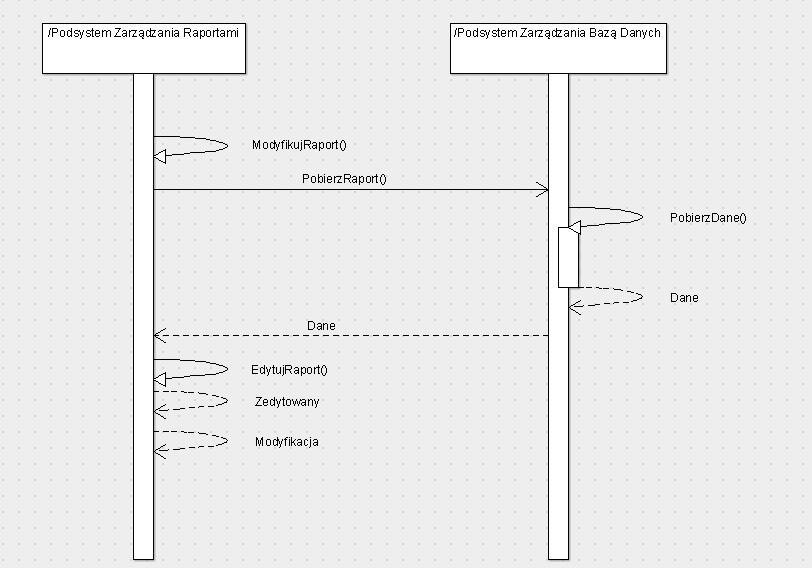
|  |  |
| --- | --- |
| Podsystem bezpieczeństwa | |
| Konto szefa bezpieczeństwa | Klasa jest odpowiedzialna za zezwolenie wejścia na teren firmy gościom ,w sytuacjach wyjątkowych otwiera bądź blokuje bramkę  atrybuty  brak  metody   * dodajKontoGoscia(Imie : String , Nazwisko : String, ID: int) : void, dodaje konto gościa wysyłając informacje o nim do bazy danych * usunKontoGoscia(ID: int) : void, usuwa konto gościa poprzez identyfikacje jego ID w bazie danych * otworzBramke() : void, zezwolenie dostępu na teren firmy * zablokujBramke() : void, blokuje bramkę w sytuacjach kryzysowych |
| Podsystem zarządzania raportami | |
| Raport | Klasa odpowiedzialna za wykonywanie zadań związanych z raportami  atrybuty  brak  metody   * pobierzRaport() : void, pobiera raport udostępniając go pracownikowi z bazy danych * modyfikujRaport() : void, umożliwia wprowadzanie zmian w raporcie po uprzednim sprawdzeniu uprawnień * sprawdzUprawnienia(ID : int) : void , określa prawa pracownika dotyczące pobranego raportu * sprawdzModyfikowanyRaport() : void, sprawdza modyfikowany raport pod względem poprawności * zapiszRaportWBazie() : void, wysyła sprawdzony raport do bazy danych * generujRaport() : void , udostępnia bazie raport, sprawdzając też czy został on wcześniej sprawdzony pod względem poprawności i ustala odpowiednie uprawnienia do przeglądania jego. |
| Posrednik | Klasa odpowiedzialna za składowanie danych  atrybuty   * raport : Object , obiekt wykorzystywany jako rekord w raportach(np. Jeżeli pobieramy raport o osobach i godzinach przepracowanych to ten obiekt składuje imie,nazwisko oraz liczbe przepracowanych godzin, czyli np Jan, Kowalski, 15.)   metody   * pobierzDanePracownika():String , pobiera dane niezbędne na rzecz raportu posiłkując się tymi dostępnymi w bazie. |
| Podsystem zarządzania bazą danych | |
| Gosc | Klasa jest odpowiedzialna za przechowywanie danych gościa  atrybuty:   * Imie : String , przechowuje imię gościa * Nazwisko : String, przechowuje nazwisko gościa * ID : int , przechowuje unikalny numer identyfikacyjny   metody  brak |
| <<interface>>  Konto | Interfejs jest wykorzystywany dla klasy Pracownik  atrybuty:  brak  metody:   * pobierzCzasPracy() : double, zlicza czas pracy jednego dnia w godzinach na podstawie algorytmu, w którym odejmuje czas wejścia z firmy od wyjścia, |
| Konto tymczasowe | Klasa odpowiedzialne za “właściwe” utworzenie konta gościa i zapisania go w bazie  atrybuty:  czasIstnienia : int , przechowuje zadeklarowany przez gościa czas w godzinach, podczas którego może wchodzić na teren firmy  metody:   * stworzKontoGoscia (date : Gosc) : String , pobiera dane osobowe gościa i tworzy indywidualne konto w bazie danych na określony czas |
| Pracownik | Klasa przechowująca dane pracownika, umożliwia tworzenie indywidualnych statystyk dotyczących jego pobytu w firmie  atrybuty:   * Imie : String, przechowuje imie pracownika * Nazwisko: String, przechowuje nazwisko pracownika * ID : int , przechowuje numer identyfikacyjny pracownika * PrzepracowaneGodziny : int, przechowuje liczbę przepracowanych godzin jednego dnia na podstawie wejść i wyjść * Nadgodziny : double, przechowuje liczbę przepracowanych nadgodzin jednego dnia * OpuszczoneGodziny: int , przechowuje liczbę opuszczonych godzin jednego dnia wyliczoną na podstawie wejść i wyjść danego dnia   metody:   * tworzStatystyke(PrzepracowaneGodziny,Nadgodziny,OpuszczoneGodziny) : void , wykonuje indywidualne statystyki dla każdego pracownika |
| Podsystem kontroli bramek | |
| Czytnik | Klasa inicjuje wejście na teren firmy  atrybuty  brak  metody:   * pobierzID() : String, czyta ID z karty RFID przy zbliżeniu karty do urządzenia * pobierzCzas() : String , pobiera czas lokalny * pobierzOdpowiedz() : String , pobiera odpowiedź weryfikacyjną * otworzBramke() : void , otwiera bramkę * pobierzDzien(data : Dni) : void , pobiera dzień, aby podstawić go za daną numeryczną * pobierzMiesiac( data : Miesiace) : void , pobiera miesiąc, aby podstawić go za daną numeryczną |
| Podsystem weryfikacji danych | |
| Weryfikator | Klasa odpowiedzialna za weryfikowanie osoby ubiegającej się o wejście/wyjście na/z teren/terenu firmy  atrybuty  brak  metody:   * pobierzDaneWchodzącego() : String, pobiera dane osoby ubiegającej się o wejście na teren firmy łącząc się z bazą danych, które są potrzebne do późniejszej weryfikacji * wyslijAlert() : void, w przypadku błędnej weryfikacji tożsamości wysyła informację do szefa bezpieczeństwa * porównajDane() : bool , metoda porównuje dane otrzymane od czytnika z tymi z bazy. False wykonuje metode wyslijAlert(). Gdy jest True to ta informacja idzie do Rejestratora dokładniej do metody pobierzDaneOsobowePotwierdzonego(). |
| Podsystem rejestracji wejścia/wyjścia | |
| Rejestrator | Klasa jest głównym przekaźnikiem potwierdzonych danych pracownika do bazy danych umożliwiających zapisanie wejścia/wyjścia  atrybuty   * liczbaWejsc() : int , zmienna pomocnicza zliczająca ilość wejść pracownika na teren firmy jednego dnia * liczbaWyjsc() : int , zmienna pomocnicza zliczająca ilość wejść pracownika na teren firmy jednego dnia   metody:   * pobierzDaneOsobowePotwierdzonego() : String , pobiera dane osobowe pracownika, który pomyślnie przeszedł weryfikację * wyslijDaneOsobowe() : void , metoda wysyła potwierdzone dane do bazy w celu odnotowania wejścia/wyjścia pracownika * sprawdzZgodnosc() : bool, metoda weryfikuje czy liczba wejść zgadza się z liczbą wyjść danego dnia.True oznacza, że pracownik nie jest na terenie firmy i można mu przeliczyć czas pobytu w danym dniu. Natomiast False, że jest na terenie firmy i nie będzie możliwe wykonanie statystyki przepracowanych godzin. |

# Diagramy sekwencji

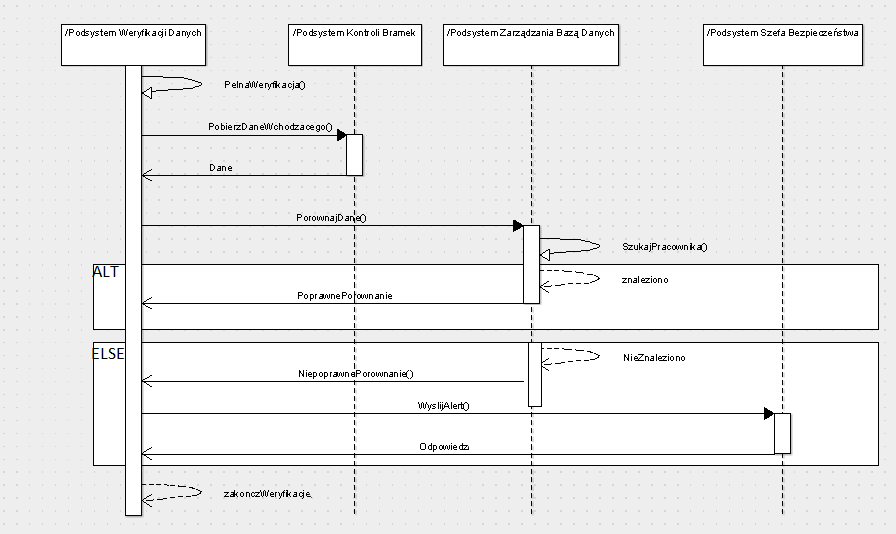
## Diagram sekwencji: “Blokada Bramki”



## Diagram sekwencji: “<Modyfikuj Raport>”

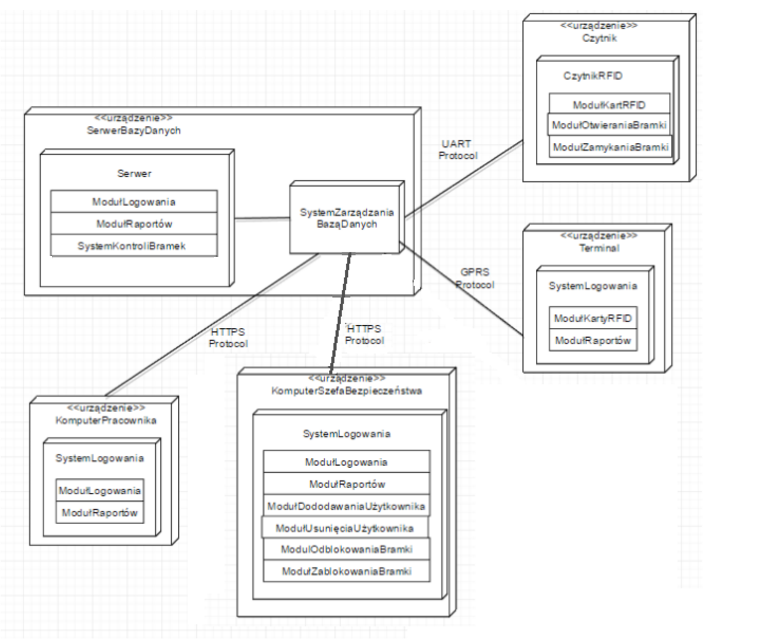


## Diagram sekwencji: “Weryfikacji pracownika”



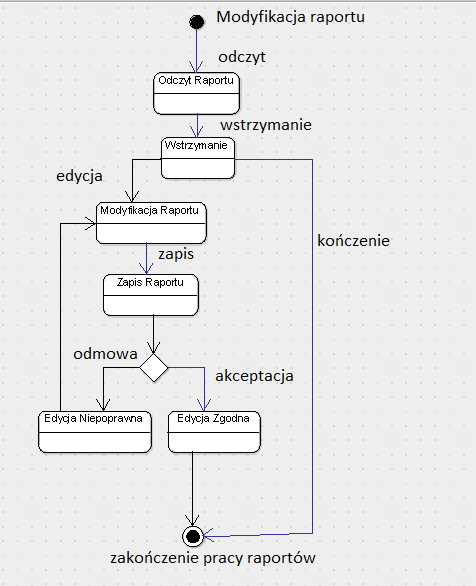
# Diagram rozmieszczenia na procesorach

## Grafika wraz z komentarzem



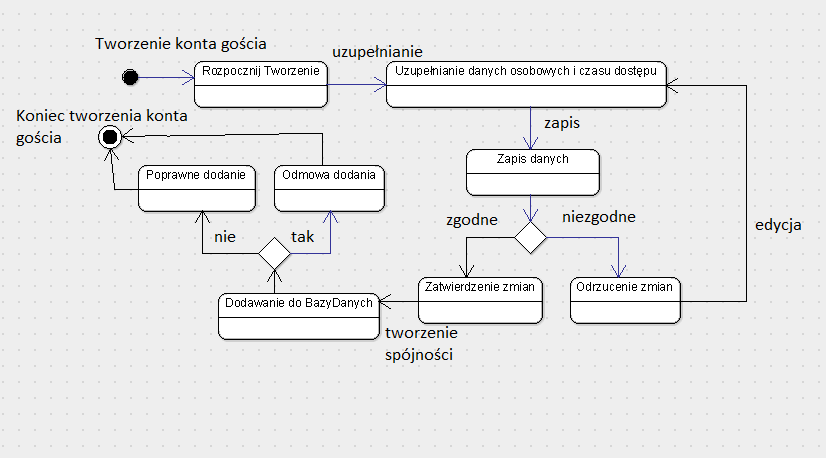
# Diagramy stanów

## Diagram stanów: “<Modyfikowanie Raportów>”

’

|  |  |
| --- | --- |
| **STAN** | **OPIS STANU** |
| Odczyt Raportu | Raport w fazie odczytu, możliwe przeglądanie. |
| Modyfikacja Raportu | Raport modyfikowany, dokonywana jest na nim edycja poszczególnych rekordów bądź atrybutów |
| Zapis Raportu | Weryfikowanie zgodności składni i logiki modyfikowanego raportu. |
| Edycja Niepoprawna | Zapis nie powiódł się przez niezgodność przy modyfikacji. Dane zostały źle wprowadzone |
| Edycja Zgodna | Dane zostały poprawnie wprowadzone, Zatwierdzenie zapisu. |
| Wstrzymanie | Oczekiwanie na wybór pracownika co do modyfikacji bądź zakończenia aplikacji. |

## Diagram stanów: “<Tworzenia konta gościa>”



|  |  |
| --- | --- |
| STAN | Opis Stanu |
| Rozpocznij Tworzenie | Tworzenie Gościa jest inicjowane przez szefa bezpieczeństwa |
| Uzupełnianie danych osobowy i czasu dostępu | Konto Gościa jest uzupełniane tożsamością i maksymalną ilością godzin w jakiej może znajdować się na terenie firmy |
| Zapis Danych | Weryfikowanie poprawności wprowadzonych danych |
| Zatwierdzenie zmian | Akceptacja zapisu |
| Odrzucenie zmian | Komunikat o niepoprawności wprowadzenia danych i odprowadzenie do ponownego wprowadzania |
| Dodawanie do Bazy Danych | Dodawanie nowoutworzonego konta do bazy pracowników |
| Poprawne Dodanie | Akceptacja dodania |
| Odmowa Dodania | Odrzucenie dodania ze względu na problemy z bazą danych (np. wykorzystana ilość możliwych gości/ Redundancja) |

# Lista katalogów i plików wchodzących w skład projektu

